

Joint Energy-based Detection and Classification of Multilingual Text Lines

Igor Milevskiy Yuri Boykov
 Computer Science Department
 University of Western Ontario, Canada
 io.milevski@gmail.com, yuri@csd.uwo.ca

Abstract. This paper proposes a new hierarchical MDL-based model for a joint detection and classification of multilingual text lines in images taken by hand-held cameras. The majority of related text detection methods assume alphabet-based writing in a single language, e.g. in Latin. They use simple clustering heuristics specific to such texts: proximity between letters within one line, larger distance between separate lines, etc. We are interested in a significantly more ambiguous problem where images combine alphabet and logographic characters from multiple languages and typographic rules vary a lot (e.g. English, Korean, and Chinese). Complexity of detecting and classifying text lines in multiple languages calls for a more principled approach based on information-theoretic principles. Our new MDL model includes data costs combining geometric errors with classification likelihoods and a hierarchical sparsity term based on *label costs*. This energy model can be efficiently minimized by *fusion moves*. We demonstrate robustness of the proposed algorithm on a large new database of multilingual text images collected in the public transit system of Seoul.

Keywords: Multi-model fitting, Multilingual text detection, Energy minimization, Minimum description length, Graph cuts, Fusion moves

1 Introduction

Over the centuries, people in different parts of the world developed various writing systems. The most common writing systems are Latin, Cyrillic, Arabic, Logographic Chinese characters, such as Hanzi, Haja and Kanji (in use in China, Korea, and Japan respectively), and alphabet based Hangul (Korea), Kitakana and Hiragana (both Japan). Most people comprehend text in one or at most two writing systems. Thus, it is extremely helpful to dedicate reading functions to a computing device. Significant progress in text detection and recognition was made in the scope of printed documents. Recently, researchers focus on more challenging tasks, such as text detection in video, web, and camera captured images [1,2,3,4], which is also the main focus of this work.

For a single line of text (e.g. in on-line handwriting recognition) or when multiple lines are easy to detect (e.g. in printed documents) the problem of text recognition can be solved with well established techniques such as the Hidden



Fig. 1: Signboard in three languages. Korean and English text lines are marked with boxes. Two Korean character form one word. There is a significant gap between them, which makes the text detection problem challenging.

Markov Model (HMM) and Recurrent Neural Networks (RNN) [5]. The input for these algorithms is an image segment corresponding to one line of text. In particular, this segment does not need to be split into isolated characters.

In the context of camera captured images, e.g. Fig.1, detection of text lines is an additional challenging problem that must be solved before image segments containing separate lines can be sent to the recognizer. Accurate partitioning of multiple text lines is critical for the success of the recognizer. This paper focuses specifically on the task of individual text line recognition, which is particularly challenging in presence of multiple languages, see Fig.2(a).

Most of the prior work on text line detection deals with scripts in Latin characters printed according to one standard set of typographical rules. In Asia, however, signboards on streets, in subway stations, railway platforms and bus terminals can have text in two or more languages at once. We aim to detect multiple lines of text from an image, e.g. Fig.1, that has characters in several languages arranged according to different typographical regulations.

For example, English typography includes letters in upper and lower cases, some characters also have descenders and ascenders. English letters can cross the mean and the baseline. One common convention is that English letters sit close to each other within each line of text, which makes text line detection relatively easy.

In contrast, Korean typographical rules make it harder to detect lines of text for two reasons. The first one is illustrated in Fig.1 where two green boxes mark Korean characters (or syllabuses) belonging to the same word. The gap between these characters is significantly greater than the distance between the first Korean character and the English line below it. Such large inter-character gaps are very common in Korean typography.

The second difficulty is that multiple Korean letters can sit on top of each other within a single text line. For example, each green box in Fig.1 corresponds to a syllabus containing 3 letters. As shown in Fig.2(c and d), it is easy to incorrectly group individual Korean letters in the same syllabus into multiple lines. Fig.2(a) shows the correct segmentation.

Similarly to Korean, Chinese characters often disperse into pieces. Thus, it could be hard to determine if one piece (a detected blob) corresponds to a character or a part of a character. Another similarity between Chinese and Korean characters distinguishing them from English letters is their fairly consistent aspect ratio, typically around one.

Multilingual text detection in camera-captured images represents a particular case of scene understanding over a small set of object classes (languages). The general scene understanding problem (e.g. on PASCAL data) is commonly solved by partitioning an image into segments of different classes, e.g.[6]. However, most methods do not separate distinct object instances within one class¹. The technical challenge of our multilingual text detection problem is that we have to segment an image into individual instances of objects (text lines) from multiple categories (languages). We formulate this as a multi-model fitting problem based on *label cost* regularization [8]. The original image is represented by a set of detected blobs, which are assigned different models/labels. Each model (text line) is described by geometric parameters² and an additional category parameter - language (Fig. 4). This category defines how the data fitting errors are computed accounting for typographical differences between the languages.

There is a lot of prior work on geometric model fitting in vision. RANSAC is the most common method for data supporting a single model. It is known for its robustness to outliers. Multi-model problems are often addressed by procedurally-defined clustering heuristics greedily maximizing inliers, e.g. Mutli-RANSAC [9], Hough space mode selection, or *j-linkage* [10]. These methods often fail on difficult examples with weak data in the presence of much noise or outliers [11]. The practical challenges of multilingual multi-line text detection motivates more powerful approaches based on optimization of a clearly defined objective function, in particular, MDL criteria [8,12]. Many previous MDL methods fit geometric models of the same class, e.g. lines. Some techniques fit a hierarchy of geometric models like *points*→*lines*→*cubes* [13] or *points*→*lines*→*vanishing points* [14]. We approach the multilingual text line detection problem by fitting geometric models (lines) from independent appearance-based categories (language) that can be interpreted as a hierarchy *blobs*→*lines*→*languages*.

Our contributions are summarized below:

- We propose a new challenging application for computer vision: multilingual text line detection over languages with different typography rules. Our database of camera-captured multilingual text images with ground truth (500 images) will be made publicly available.

¹ Some methods use object detectors to count object instances [7]

² Due to perspective distortion we represent *base*- and *mean lines*.



Fig. 2: Examples of ambiguities in multilingual text line detection. a) Ground truth with Korean, English, Chinese characters and digits. b) Detected blobs. Green boxes come from connected component analysis. Red boxes are placed using heuristics to cover potential Korean characters c) Minimization of geometric errors alone leads to incorrect grouping. d-e) Classification ambiguities: d) two Korean characters on top, two digits on the bottom. e) two Korean characters are seen. Both d) and e) are equally good for the classifier.

- We propose a novel hierarchical MDL energy for multilingual text detection (1). Our sparsity-based (label cost) regularization is applicable to a wider range of typographies compared to smoothness-based approaches [15,16] assuming proximity between letters. Our energy can be efficiently optimized by fusion moves [14].
- In contrast to many standard techniques for scene understanding, our method simultaneously segments an image into multiple classes (languages) and into individual object instances (text lines) within each class by observing geometric errors and classification likelihood.
- Unlike previous text detection algorithms assuming a single language with no grouping ambiguities as in Fig.2, we solve a much harder clustering problem that can not be addressed with standard simple heuristics. Instead, we have to formulate and optimize an information theoretic criteria to resolve all ambiguities in the proposed ill-posed problem. Our MDL approach is closely related to semantic segmentation. It has Bayesian interpretations [17,18,12,19].

The rest of the paper is organized as follows. Sec.2 discussed related prior work on text detection. Sec.3 presents our new hierarchical MDL formulation of the multilingual generalization of this problem. Our camera-captured text image database and experimental evaluation of our algorithm is presented in Sec.4.

2 Related work

Interest in text detection has increased since 2003 when the ICDAR database and competition was formed. After that, ICDAR 2011, Street View Text(SVT) and private publicly unavailable databases were collected. All mentioned databases include text in Latin script. A number of text detection algorithms were proposed and evaluated on these image collections.

In general, a text detection algorithm combines the following parts: text candidate detection, text candidate filtering and line fitting. For a given input image the algorithm produces a set of rectangles either horizon-oriented or rotated.

There are three major groups of text candidate detection methods: sliding window [2,15,20,21,22,23], edge based [16,24,25], and color based [3,26] algorithms.

Sliding window algorithms, originally proposed for face detection, denote an exhaustive search. Features are computed for each position and scale of the sliding window.

Edge based methods retrieve an edge map (Sobel, Canny, Laplace) and then perform connected component (CC) analysis and outputs blobs. Moreover, Stroke width transform (SWT) [1,4] also aims to find blobs that have consistent width of stroke. Color based methods, such as MSER and ER (inspired by MSER) assume that a text character's color is homogeneous. MSER was used by the "ICDAR 2011 robust reading competition" winner.

After text candidates are detected non-text blobs must be filtered out. The decision whether a blob represents text is done by classification. Popular classifiers are: support vector machine (SVM) [24,26,27], AdaBoost [28,29] or their cascades. Popular features for classification are: color based (histogram of intensities, moment of intensity) edge based (histogram of oriented gradients (HOG), Gabor filter) geometric features (width, height, aspect ration, number of holes, convex hull, area of background/foreground).

Single text blobs must then be aggregated into text lines. Older approaches are based on the Hough transform algorithm [30,31]. More recent algorithms combine neighbours of blobs into pairs and then fulfill clustering in N-dimensional space, where the following dimensions are in use: stroke width, orientation of a pair, and geometric size of blobs.

Text candidate filtering and text line detection could be done consequentially with complex approaches based on a Markov random field (MRF) [2,22] or a Conditional Random Field (CRF) [32] and algorithms based on minimal spanning trees.

In [30,33] the authors assume that text is located on a signboard. They try to locate physical edges and corners of that signboard and perform signboard rectification.

3 Algorithm

The general work flow of our algorithm is shown in Fig. 3. First, in the input image blobs or text candidates are detected. Next, in addition to geometric prop-

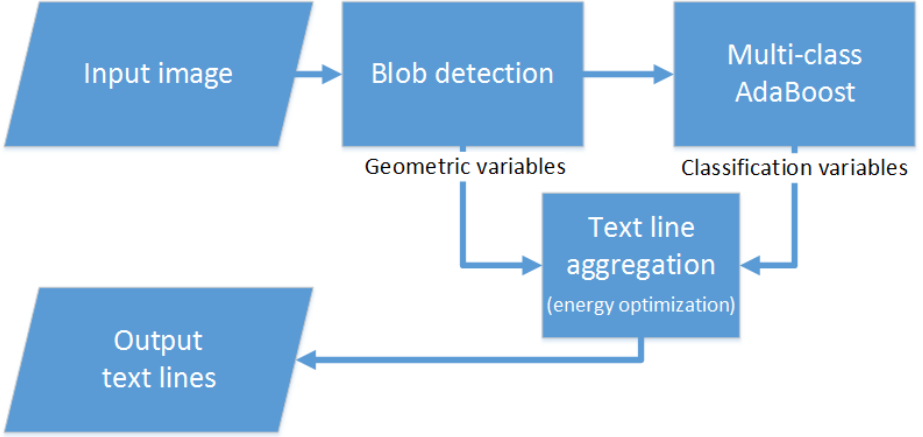


Fig. 3: Energy-based algorithm for joint text line detection and classification.

erties each blob gets an likelihood to belong to one of the five categories (English, Korean, Chinese, Digit, Non-text). The likelihood is provided by an AdaBoost classifier. Finally, energy-based algorithm groups blobs into text lines. This can be seen as text candidate clustering in multidimensional space of geometric and classification values.

Let us recall the challenges of the multilingual text detection problem:

- The number of languages and the number of text lines is not known.
- A text candidate can occupy a whole text string’s height or a part of that sting. In other words there is an ambiguity of grouping text candidates of an Asian script (See Fig 2). Therefore, a sequential classification-aggregation algorithm i.e. [32] is not applicable to our problem.
- Smoothness term, standard to MRF, can not be used since it makes sense only for European scripts (See Fig 1).

The MDL principle suggest that the solution of the problem is the smallest set of text lines which fits text candidates with the smallest geometric and classification errors. The solution must have text lines in only a few languages.

Our joint text detection and classification approach is based on the hierarchical energy minimization:

$$E(\mathbf{l}, \boldsymbol{\theta}) = \sum_{i \in \mathcal{I}} D_i(l_i | \boldsymbol{\theta}) + \sum_{j \in \mathcal{L}} C_j \cdot [\exists l_i = j] + \sum_{v \in \mathcal{V}} C_v \cdot [\exists v_{l_i} = v], \quad (1)$$

where \mathcal{I} is the set of all text candidates. Text candidate detection is described in Subsec. 3.1. \mathcal{L} is the space of all possible text line models. \mathcal{V} is the space of all possible languages. The vector $\boldsymbol{\theta}$ contains text line model parameters:

$$\boldsymbol{\theta} = \{\theta_j | j \in \mathcal{L}\}. \quad (2)$$

Text line model parameters θ_j are a language mark and parameters of base and mean lines (See Fig. 4). The lines of a model are not necessarily parallel. The model can describe perspectively distorted or arbitrary rotated text lines in either English, Chinese or Korean. The vector \mathbf{l} contains indexes of text line models assigned to each text candidate i :

$$\mathbf{l} = \{l_i \in \mathcal{L} | i \in \mathcal{I}\}. \quad (3)$$

$D_i(j|\theta)$, the data term, tells how well a particular text candidate i fits text line model j :

$$D_i(j|\theta) = \overbrace{-\ln \Pr(i|\text{lang}_j)}^{\text{classification}} + \overbrace{\text{dist}_{\text{lang}(j)}(i, \text{lines}_j)}^{\text{geometry}}. \quad (4)$$

The data term consist of two values: the classification likelihood (Subsec. 3.3) and the geometric error (Subsec. 3.2). The geometric error is scaled according to the language of the model. The outlier cost $D_i(\emptyset) = \text{const}$ is paid if the blob is not associated with any text line model. Outlier model works like a non-text blob collector.

C_j penalizes number of text line models in the solution. The penalty C_j is counted if there is at least one data point i assigned to the text line model j . C_v penalizes number of languages in the solution. The penalty C_v is counted if there is at least one data point i assigned to the language v .

The solution of the optimization problem (1) is the set of text line model parameters θ , and labeling \mathbf{l} . A labeling maps text candidates into the text lines.

The energy (1) describes the following hierarchical levels: *blobs* \rightarrow *text lines* \rightarrow *languages*.

3.1 Blob detection and proposal

We use an edge based blob detector to produce text candidates. At first, an input image is converted to gray-scale and downsized for performance reasons. Next, horizontal and vertical Sobel filters are applied. Two resulting images are combined. The result represents an edge map. Finally, connected component analysis run resulting character candidates (blobs).

Unlike a color based blob algorithm, an edge based method does not have the foreground-background ambiguity when algorithm must be run twice for dark text on bright background or vice versa.

More complex blob detectors were considered. Chinese symbols are very compact. They have high density of strokes per unit of area. In such conditions, SWT fails to detect Chinese characters well.

At this point there are Korean characters that are over-segmented (Fig. 2 b). It is necessary to propose more text candidates to cover such characters. Two or three blobs produce a new blob if they stand close to each other, cover resulting area well and the resulting aspect ratio is close to one.

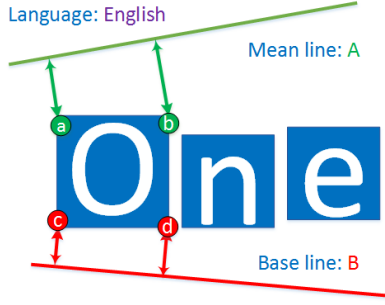


Fig. 4: Geometric errors.

3.2 Geometric errors

A text candidate i is more likely to be a part of a text line j if it sits close to its base and mean lines. The geometric part of the data term (4) includes the sum of Euclidean distances between four corners of a text candidate's box and the base and the mean line of a model (Fig. 4)

$$\begin{aligned} dist_{lang(j)}(i, lines_j) = & \quad (5) \\ & \frac{K_{lang(j)}}{Z_j} \cdot (||A_j - a_i|| + \\ & ||A_j - b_i|| + \\ & ||B_j - c_i|| + \\ & ||B_j - d_i||), \end{aligned}$$

where $K_{lang(j)}$ is a language dependent scale. There are ascender and descender characters in English alphabet. In contrast to English, Korean and Chinese characters have roughly same height. We allow larger tolerance for geometric error between the text candidates and English lines, and smaller tolerance for geometric errors between the text candidates and Chinese or Korean text lines. Z is a constant that depends on the height of text line j .

3.3 Classifier and features

AdaBoost is a machine learning algorithm that builds a strong classifier out of weak ones. AdaBoost have become a popular choice for text vs. not-text recognition. In our problem we use a multi-class AdaBoost, which is trained on the following categories: English, Korean, Chinese, Digit, Non-Text. Decision trees are commonly used as a weak classifier. It is possible to scale the classifier according to the complexity of the problem by adjusting the depth and the quantity of the trees.

A blob that is marked as “text” must cover a text line’s height completely. If it covers two or more lines (case of under-segmentation), or only a portion of a text line’s height (case of over-segmentation) then it is treated as non-text.

The following groups of features are used for AdaBoost classifier training: colour, gradient and geometry based features.

A bounding box that contains text has normally two distinctive colors - one for foreground, one for background. Non-text blobs do not have such consistency. Color features are useful for text vs. non-text classification. Histogram of intensities descriptor is computed as in [28]. Text of a different language could be diversified according to stroke density and number of strokes in a particular direction. We use histogram of oriented gradients (HOG) and Gabor filter to describe such properties. Geometry based features are blob’s width, height and width-to-height ratio.

3.4 Energy optimization details

Minimization of energy (1) is an NP-hard mixed optimization problem combining integer labeling variables and real-valued model parameters. Nevertheless, there are approximation algorithms [8,11,14], which guarantee some optimality bounds and, in practice, give good approximate solutions.

We adopt *PEARL*, see Algorithm 1, introduced by Isack et al. [11]. The first step produces a finite set of models by randomly sampling data points as in RANSAC. Then, block coordinate descend (BCD) procedure minimizes energy (1) iterating two steps: (*AssignModels*) selecting an optimal labeling by optimizing over discrete variables \mathbf{l} and (*RefitModels*) optimizing real-valued model parameters θ .

Algorithm 1: PEARL // ENERGY-BASED MODEL FITTING ALGORITHM

Input: *data_points* // blobs, etc.

Output: *labeling* // data_points-to-model map
models // text lines

```

1 pool  $\leftarrow$  RandomSampleModels(data_points);
2 for  $i \leftarrow 0$  to max_iterations do
3   labeling  $\leftarrow$  AssignModels(pool, data_points);
4   pool  $\leftarrow$  RefitModels(pool, data_points, labeling);
5 models  $\leftarrow$  pool
6 return [models, labeling];
```

We compute our initial set of models as follows. First, a neighbourhood map is build by Delaunay triangulation. Next, blobs that have a common edge produce a model. Unlike k-nearest neighbour, such an approach enables a pair of blobs to produce a text line model, regardless of the distance between the blobs.

AssignModels routine assigns each blob to a model (label) from the pool based on an optimal labeling with respect to hierarchical MDL energy (1) assuming fixed model parameters θ . The corresponding optimization algorithm is described in details below. *RefitModels* step re-estimates optimal model parameters θ assuming fixed labeling \mathbf{l} . In this case, only the data term of energy (1) depends on parameters θ . Thus, optimal geometric parameters for each model should be independently computed for each set of inliers, that is, to blobs assigned the same model (label). The top and the bottom lines of a single model are refit by linear least squares (LS).

Labeling Optimization: Procedure *AssignModels* computes an optimal discrete labeling \mathbf{l} for energy (1) with fixed parameters θ using the standard *fusion move* or *optimized crossover* framework [34,14], as summarized in Algorithm 2. The main idea of *fusion moves* is to iteratively merge a sequence of proposed solutions into one optimal labeling. While similar to α -expansion [35], the main difference is that proposals do not have to be constant labelings.

First, the initial labeling is computed: each blob is either assigned to the outlier model or to the first model in the pool based on the data term (4). Then, at each iteration, the current solution l^0 is “fused” with some new labeling proposal l^1 that was also formed by combining a model from the pool and the outlier model.

Algorithm 2: ASSIGNMODELS

Input: *blobs* // text candidates
models // text lines

Output: *labeling* // blob-to-model map

```

1  $l_0 \leftarrow \text{MakeLabeling}(\emptyset, \text{pool}[0], \text{blobs});$ 
2 for  $i \leftarrow 1$  to  $n\_models$  do
3    $l^1 \leftarrow \text{MakeLabeling}(\emptyset, \text{pool}[i], \text{blobs});$ 
4    $l^0 \leftarrow \text{FuseLabeling}(l^0, l^1);$ 
5  $labeling \leftarrow l^0$ 
6 return  $labeling;$ 
```

The *fusion move* is a binary optimization procedure for merging two solutions l^1 and l^0 combining their best properties in one new labeling \mathbf{l} . This is analogous to crossover in genetic algorithms. There are many different ways to fuse two solutions. Any fusion result \mathbf{l} is uniquely defined by a binary vector \mathbf{x} specifying which of the components are borrowed from solution l^1 and which are from l^0 . In particular,

$$\mathbf{l}(\mathbf{x}) = \mathbf{x} \circ \mathbf{l}^1 + (1 - \mathbf{x}) \circ \mathbf{l}^0, \quad (6)$$

where \circ represents a point-wise product of two vectors, a.k.a. *Hadamard* product. In other words, each component of a fused labeling is defined by binary variable $x_i \in \{0, 1\}$ as $l_i = x_i \cdot l_i^1 + (1 - x_i) \cdot l_i^0$ for every text candidate $i \in \mathcal{I}$.

The goal of a *fusion move* is to find an optimal \mathbf{x} giving the best possible crossover solution $\mathbf{l}(\mathbf{x})$ with respect to energy (1). When model parameters $\boldsymbol{\theta}$ are fixed, (1) defines the energy for $\mathbf{l}(\mathbf{x})$ as a function of \mathbf{x}

$$\begin{aligned} E(\mathbf{l}(\mathbf{x})) = & \sum_{i \in \mathcal{I}} D_i^0 + (D_i^1 - D_i^0)x_i + \\ & \sum_{j \in \mathcal{L}} C_j \cdot (1 - X_{P_j^0} \bar{X}_{P_j^1}) + \\ & \sum_{v \in \mathcal{V}} C_v \cdot (1 - X_{P_v^0} \bar{X}_{P_v^1}), \end{aligned} \quad (7)$$

where \mathcal{I} is the set of all detected text candidates, \mathcal{L} is the set of proposed text line models, \mathcal{V} is the set of languages. The data term is defined as

$$\begin{aligned} D_i^0 &= D_i(l_i^0 | \boldsymbol{\theta}) \\ D_i^1 &= D_i(l_i^1 | \boldsymbol{\theta}). \end{aligned} \quad (8)$$

Constants C_j and C_v are the costs of each text line model and each language as in energy (1), and

$$\begin{aligned} X_{P_j^0} &= \prod_{p \in P_j^0} x_p \\ \bar{X}_{P_j^1} &= \prod_{p \in P_j^1} \bar{x}_p, \end{aligned} \quad (9)$$

where $\bar{x}_p = 1 - x_p$ and $P_j^k = \{i \in \mathcal{I} | l_i^k = j\}$.

Note that $X_{P_j^0} \bar{X}_{P_j^1}$ is a boolean expression that turns “on” and “off” text line costs in (7). The switch is “on” if there is at least one text candidate that supports text line model j and the cost of the model is paid. Otherwise, the switch is “off” and the penalty is not paid. $X_{P_v^0} \bar{X}_{P_v^1}$ is defined similarly to (9).

Binary energy (7) includes unary potentials and high-order terms that came from the hierarchical label costs in energy (1). Similar high-order potentials were studied by DeLong et al. [14] in the context of hierarchical clustering. We find a globally optimal *fusion move* minimizing (7) using the same graph cut construction as proposed in [14].

4 Evaluation

Our database of signboards from the Seoul subway consist of 500 images taken with a Samsung Galaxy SII, Galaxy S, Apple iPhone 3GS, Canon EOS 450D. 400 images were used for training and the remaining 100 for testing of both the AdaBoost classifier and the whole algorithm evaluation.

We want to test all three modules of the algorithm in an isolated fashion. However, it is not straightforward to check how well detected blobs cover text on

an image before the text lines are extracted. Instead, we compare the final line segments detected in one image to two ground truths - the first containing blobs detected programmatically by the blob detection module (Artificial base), the second containing blobs placed by a human (Real base). The difference between the results of these two experiments, see Table 1, reflects the quality of the blob detector.

Ground truth	Recall (%)	Precision (%)	F (%)
Real base	71	84	76
Artificial base	76	84	79

Table 1: Text line detection accuracy.

The confusion matrix for text candidate classification by AdaBoost is shown in Table 2. The average recognition rate is 83.35 %. The examples of text line detection is shown in Fig. 5.

Actual \ Predicted					
	English	Korean	Chinese	Digit	NonText
English	902	10	16	10	45
Korean	52	815	63	6	40
Chinese	27	14	172	0	11
Digit	19	1	0	121	5
NonText	482	305	188	144	5278

Table 2: Confusion matrix for classification.

5 Conclusions

In this work we introduce a challenging new problem of the multilingual multi-line text detection. We formulate the problem as a hierarchical MDL energy optimization and demonstrated that a fusion based method efficiently obtains good quality solutions for this energy. We obtain very promising results on our large database of images from the subway of metropolitan area of Seoul that we plan to make public for other researchers in computer vision. Our experiments show that the bottleneck of our algorithm is the AdaBoost classifier performance, which could be enhanced. Extending our current hierarchy *blobs*→*lines*→*languages* into *blobs*→*characters*→*lines*→*languages* could further improve the results.



(a)



(b)



(c)



(d)



(e)



(f)

Fig. 5: Results of text line detection. Triplet of lines in one color shows one line. Circles mark centers of text candidates. Blobs assigned to the outlier label are hidden for readability.

References

1. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. (2010) 2963–2970
2. Mishra, A., Alahari, K., Jawahar, C.: Top-down and bottom-up cues for scene text recognition. (2012) 2687–2694
3. Neumann, L., Matas, J.: Real-time scene text localization and recognition. (2012) 3538–3545
4. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. (2012) 1083–1090
5. Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **31**(5) (2009) 855–868
6. Ladicky, L., Russell, C., Kohli, P., Torr, P.H.S.: Graph cut based inference with co-occurrence statistics. (2010) 239–253
7. Ladický, L., Sturges, P., Alahari, K., Russell, C., Torr, P.H.S.: What, where and how many? combining object detectors and crfs. (2010) 424–437
8. DeLong, A., Osokin, A., Isack, H., Boykov, Y.: Fast approximate energy minimization with label costs. (2010) 2173–2180
9. Zuliani, M., Kenney, C., Manjunath, B.: The multiransac algorithm and its application to detect planar homographies. **3** (2005) III–153–6
10. Toldo, R., Fusiello, A.: Robust multiple structures estimation with j-linkage. (2008) 537–547
11. Isack, H., Boykov, Y.: Energy-based geometric multi-model fitting. *Int. J. Comput. Vision* **97**(2) (April 2012) 123–147
12. Torr, P.H.S.: Geometric motion segmentation and model selection. *Phil. Trans. Royal Society of London A* **356** (1998) 1321–1340
13. Porway, J., Wang, K., Yao, B., Zhu, S.C.: A hierarchical and contextual model for aerial image understanding. (2008) 1–8
14. DeLong, A., Veksler, O., Osokin, A., Boykov, Y.: Minimizing sparse high-order energies by submodular vertex-cover. (2012) 971–979
15. Pan, Y.F., Hou, X., Liu, C.L.: A hybrid approach to detect and localize texts in natural scene images. *Image Processing, IEEE Transactions on* **20**(3) (2011) 800–813
16. Shivakumara, P., Phan, T.Q., Tan, C.: A gradient difference based technique for video text detection. (2009) 156–160
17. Leclerc, Y.: Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision* **3**(1) (1989) 73–102
18. Zhu, S.C., Yuille, A.: Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **18**(9) (Sep 1996) 884–900
19. DeLong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. *Int. J. Comput. Vision* **96**(1) (January 2012) 1–27
20. Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D., Ng, A.: Text detection and character recognition in scene images with unsupervised feature learning. (2011) 440–445
21. Karaoglu, S., van Gemert, J., Gevers, T.: Object reading: Text recognition for object recognition. **7585** (2012) 456–465
22. Pan, Y.F., Zhu, Y., Sun, J., Naoi, S.: Improving scene text detection by scale-adaptive segmentation and weighted crf verification. (2011) 759–763

23. Zhao, Y., Lu, T., Liao, W.: A robust color-independent text detection method from complex videos. (2011) 374–378
24. Petter, M., Fragoso, V., Turk, M., Baur, C.: Automatic text detection for mobile augmented reality translation. (2011) 48–55
25. Phan, T.Q., Shivakumara, P., Tan, C.: A laplacian method for video text detection. (2009) 66–70
26. Neumann, L., Matas, J.: Text localization in real-world images using efficiently pruned exhaustive search. (2011) 687–691
27. Minetto, R., Thome, N., Cord, M., Stolfi, J., Precioso, F., Guyomard, J., Leite, N.J.: Text detection and recognition in urban scenes. (2011) 227–234
28. Lee, J.J., Lee, P.H., Lee, S.W., Yuille, A., Koch, C.: Adaboost for text detection in natural scene. (2011) 429–434
29. Shahab, A., Shafait, F., Dengel, A.: Icdar 2011 robust reading competition challenge 2: Reading text in scene images. (2011) 1491–1496
30. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic detection and recognition of signs from natural scenes. *Image Processing, IEEE Transactions on* **13**(1) (2004) 87–99
31. Shiku, O., Kawasue, K., Nakamura, A.: A method for character string extraction using local and global segment crowdedness. **2** (1998) 1077–1080 vol.2
32. Pan, Y.F., Hou, X., Liu, C.L.: Text localization in natural scene images based on conditional random field. (2009) 6–10
33. Cambra, A.B., Murillo, A.: Towards robust and efficient text sign reading from a mobile phone. (2011) 64–71
34. Lempitsky, V., Rother, C., Roth, S., Blake, A.: Fusion moves for markov random field optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **32**(8) (Aug 2010) 1392–1405
35. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **26**(9) (Sept 2004) 1124–1137